EXHIBIT C

Certificate of Registration

This Certificate issued under the seal of the Copyright Office in accordance with title 17, *United States Code*, attests that registration has been made for the work identified below. The information on this certificate has been made a part of the Copyright Office records.

United States Register of Copyrights and Director

Registration Number
TX 8-966-581
Effective Date of Registration:
April 28, 2021
Registration Decision Date:
May 27, 2021

Title

Title of Work: Vulcanize - Wireline Registry Mechanisms

Completion/Publication

Year of Completion: 2018
Date of 1st Publication: November 10, 2018
Nation of 1st Publication: United States

Author

- **Author:** Vulcanize, Inc.
Author Created: text
Work made for hire: Yes
Citizen of: United States

Copyright Claimant

Copyright Claimant: Vulcanize, Inc.
244 Fifth Avenue, #D281, New York, NY, 10001

Rights and Permissions

Organization Name: Butzel Long
Name: Jennifer Ann Dukarski
Email: dukarski@butzel.com
Telephone: (734)213-3427
Address: 301 East Liberty
Suite 500
Ann Arbor, MI 48104 United States

Certification

Name: Jennifer Dukarski
Date: April 28, 021

Vulcanize Wireline Mechanism

Rick Dudley and Eric Olszewski

November 6, 2019

Wireline Mechanism

Draft Q3-2019

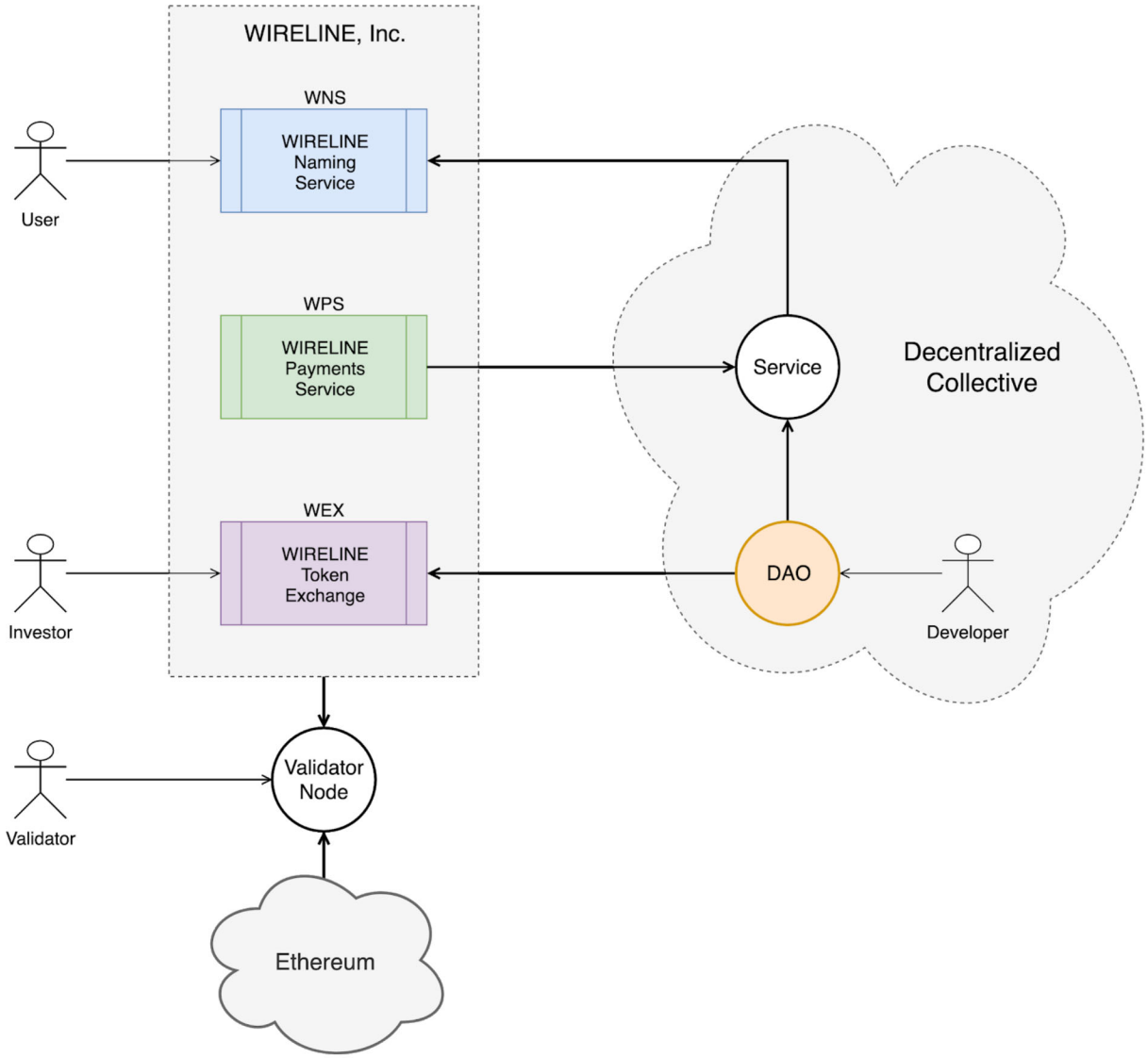
Overview

The WIRELINE network is an ecosystem for decentralized open source applications and services. It consists of infrastructural services provided by Wireline, Inc. and an owner-operated DAO--The Decentralized Collective (DxCollective)--which is a cooperative made up of independent software developers across the world.

The ecosystem consists of four principal actors:

Developers	Independent software developers create and operate open source software that runs on the network.
User	Users use software and services.
Investors	Investors provide capital for developers. They sponsor the development of new projects
Validators	Validators run blockchain nodes that secure network services.

The diagram below illustrates the relationship between ecosystem participants.



Wireline sidechain

1. WIRE has intrinsic value:
    a. Token sale is WIRE in exchange for on-chain assets (marked to USD price at the time of sale):
        i. ETH
        ii. A collection of stablecoins
        iii. BTC?
    b. Validators are staked in WIRE
    c. Validators are rewarded in WIRE
    d. Fees are paid in WIRE
    e. Perpetual real-time atomic swap market for WIRE exists in WEX
2. Escrow service described in WPN paper and shared with WEX
    a. Escrows ERC-20s issues:
        i. WUSD
        ii. WIRE
        iii. Vouchers for ERC-20 assets (which can be used for trading on WEX)
3. WPN
    a. Escrows ERC-20 WUSD, grants WPN WUSD Vouchers
    b. Opens state channels used to pay for services
    c. Allows for simple payments between users
4. WEX
    a. Auctions for the atomic swapping of assets taken as collateral for WUSD
        i. ETH
        ii. DAI
    b. Allows for the sale of STX in exchange for WIRE
        i. Uses configurable price curves, which can be used to generate liquidity proofs
    c. Provides atomic swap based markets required for the system to function:
        i. WIRE <-> WUSD
        ii. ERC-20s backing WIRE
    d. Fees are taken in both currencies transacted and placed into the WIRE backing pool.
    e. WIRE is a derivative of its backing assets
    f. STXs are derivatives of WIRE
    g. (Perpetual sales have interesting dynamics related to bonding periods)
    h. TODO: explain the asset pool, it was omitted from the WPN paper
    i. TODO: explain provably fair matching and its relationship to derivatives
5. WNS
    a. BFT graph of records which address different types of resources available on the wireline network.
    b. This is how end-user apps find resources.
    c. Collection of IPLD objects

d. We need something like the merkleroot of the head published revision of an application
    i. By using IFFS, we can reference the git commit histories
e. Supports IPFS
f. Supports generating proofs/record history
g. Clients can query quorums of nodes and get receipts
    i. Receipts may be closed WPN channels
h. Reputation of registry nodes is private but includes public information
    i. How many failed WPN channels

These all share a federated WIRE-bonded sidechain. We have to be sure to “slash in the right direction” There are times where we want to burn the WIRE but also remove some of the assets from the pool, and other times when we don’t. There are a lot of different forms of punishment in the system that need to be thought of clearly.
WNS for service to service payments
WPN for onboarding all users
WEX to allow investors to fund the development of services and then take profits.

FAQ

1. What is the relationship between Wireline and the DxCollective?

Wireline Naming Service Spec

- Underlying tech: Rick recommends sticking with Cosmos
- Mechanism design:
    - Querying: can be paid and metered.
        - We want to have a free tier that won't affect an average end-user — e.g. querying for apps/bots should from Launchpad should be free
        - Complexity and value return size affect price, paid in WIRE, priced in bytes/WIRE
        - P2P transaction between requester and provider (node)
            - Node advertises price for each type of query if user accepts they open a state channel and pay (using WPS).
            - It would be a streaming bytes-for-cents payment protocol.
            - Bad behavior on the part of the node would available from viewing the settlement of the state channel
            - Bad behavior on the part of the user would result in blacklisting on a single node. Nodes may choose to share blacklists. Bonding clients here is possible, but almost surely unnecessary.
    - Create/update records
        - Creating records is more expensive than updating
        - Send a transaction to all validators to be included in a block
            - Transactions have basefee (burn) and a processing fee which goes to the proposer of the block the transaction is sealed in.
        - Similar model as querying: node advertises price for each type of query in streaming bytes/WIRE model.
    - Zones (Human readable names)
        - Bound to entity identity
        - Paid in advance to the wireline validator set with bond in WIRE:
            - Generic Transaction burn (per byte)
            - Standard processing fee (per byte)
            - A years payment in advance (target $5 of WIRE per year, up to 100 years)
        - Bond can be refilled at any time up to some upper limit (100 years or something)
        - User can withdraw their bond.
            - There is no inflationary reward on name subscriptions, so the incentive is to keep the name at the $5 minimum.
        - Once bond expires, for a given time period:

        - if two or more entities want it
            - name is auctioned
        - If only original entity wants it
            - Renewed with extra fee
        - If only one new entity wants it
            - Same cost as registering new
            - Redirects are supported for old entity
        - If no one wants it:
            - Name is released with no redirect.
    - Need to do work on pricing/governance for names (e.g. shorter names more expensive)
- Why we aren't going to implement any reputation system for the Launchpad demo:
    - How do we know the relative quality of two entities claiming to provide identical service?
        - We can rely on non-monetary statements from trusted parties
            - Review services like websites or user submitted statements.
            - Easy to implement, easy to game.
        - We can look at our past transactions with a service and derive a reputation
            - Reputation is a bilateral metric between two parties
        - We can imagine the service that has more "value staked on it" has more trustworthy metrics (This is basically a TCR)
            - This would be WIRE staked by anyone on a given service.
            - They would share in the profits generated by the service
            - The stake would be slashed, if a user of the service could prove to the validators that the service failed to meet its own service level agreements in some way. (Payment channels are the easiest example of this.)
        - We can look at public transaction history
            - A service may choose to reveal its payment history to demonstrate its quality
                - Not sybil resistant
                - Service would only show positive actions
                - Users would need to be able to reveal their activities with a service, this has complex privacy implications.
- Next steps:

-

- Describe The Wireline Network
    - Core services (Primarily provided to end-users)
        - Name
        - Payments
        - Exchange
    - Who are validators
        - Validators are entities that are bonded in WIRE and running the core services on behalf of users that pay in WIRE. The network can support an arbitrary large (e.g., thousands) number of validators. Based on other validator-based networks (e.g. COSMOS, EOS) we anticipate something closer to 20-30 bonded validators initially on a mainnet.
        - There are a number of criteria which can be used to select which validators can bond and become block producers. Most importantly they need to provide assets which can be slashed when they fail at their consensus-assigned tasks.
        - Validators will run the same open source client, initially developed and provided by Wireline, and provide all core services.
    - What state is being replicated across validators?
        - Hashes from multiple existing networks such as other blockchains or Wireline Parties
        - Data
            - Via Cosmos IBC
        - Value
            - Via Lightning HTLCs across multiple blockchains
            - Issuing vouchers
            - Running an exchange
    - Why are they trusted
    - How are validators governed?
    - How are they rewarded
    - how are they punished

Additional notes:
1. Pricing wire/byte

1 kb should be about 1 penny

“We have a price oracle in the system” (because we’re running an exchange) -> maybe this should be a stablecoin; otherwise we should reprice dynamically using the price oracle

Continual sale will provide price stability (Vlad/Rick proposal)

2. Pricing for naming/rent
    Q1 - how do we relate to DNS? Are they just machine names?
        “We have to do that”
    Q2 - how do we address squatting?
    Ethereum ran a bunch of tests on pricing
    We probably want to end up with a number between $3-10/year
    We should stake/bond per WIRE, pay in WUSD
    Rick would do multi-coin.
    “Price it aggressively so that people don’t put movies in there”
3. Queries/reads
    Queries happen though a state channel
    Query will be cryptographically signed with proof; you can verify elsewhere. There may be slashing of the service if the result is incorrect.
4. Governance
    TBD
5. Macro/emission
    TBD. How much was pre-sold, how much will be sold in the future, etc.
    E.g. this should represent X% of tokens for the next Y years. (Say 50% for the next 5 years).